CLAIMS

1. A computer system, comprising:

a compiler receiving higher-level code and outputting lower-level code to enable a processor to simultaneously process multiple multi-bit data elements in a single register, the logic of the lower-level code including:

establishing at least first and second signed, multi-bit data elements in at least a first register; and

simultaneously processing the elements.

2. The computer system of Claim 1, wherein the compiler accesses at least one of: a compiler directive, a flag, or a configuration file, to decide when to make elements independent of each other.

3. The computer system of Claim 1, wherein a first element is provided from a first data set and a second element is provided from a second data set different than the first.

4. The computer system of Claim 1, wherein the compiler allocates a respective output precision in a register for each data element to be processed in the register during a single cycle.

5. The computer system of Claim 1, wherein the compiler receives instructions not to compile a predetermined portion of code received by the compiler.

6. The computer system of Claim 1, wherein an output precision or an input precision is defined by means of a compiler directive, or a configuration file, or a variable definition.

7. A computer program device comprising:

a computer program storage device readable by a digital processing apparatus; and

a compiler program on the program storage device and including instructions executable by the digital processing apparatus for performing method acts for outputting lower-level code to process multi-bit, signed data elements, the lower-level code comprising:

computer readable code means for packing at least first and second data elements into a single register; and

computer readable code means for processing the elements simultaneously.

8. The computer program device of Claim 7, further comprising:

flag means indicating whether a precision should be checked in at least one cycle.

9. The computer program device of Claim 7, further comprising:

compiler directive means for defining an input precision.

10. The computer program device of Claim 7, further comprising:

compiler directive means for defining multiple data sources of respective data elements to be packed into a common register and operated on by an algorithm simultaneously with each other.

11. A method, comprising:

defining at least one compiler directive, instructions, or configuration file for a compiler defining at least one of:

an input precision for at least one data element; and

multiple data sources of respective data elements to be packed into a common register and operated on by an algorithm simultaneously with each other.

12. The method of Claim 11, wherein the compiler determines first and second precisions to be allocated in a single register to hold respective first and second signed data elements, and the compiler generates a lower-level code from a higher level code to undertake method acts comprising:

packing the elements into the register; and

operating on the elements.

13. The method of Claim 12, wherein the register sends plural data elements simultaneously to at least one computational subsystem.

14. The method of Claim 13, wherein the operation is a multiplication by a constant or by a variable of known precision, or an addition, or a shift-left logical, or a subtraction, or a bitwise AND, or a bitwise OR.

15. The method of Claim 14, wherein the elements are independent of each other as defined by the compiler directive or configuration file.

16. The method of Claim 15, wherein the first element is provided from a first data set and the second element is provided from a second data set different than the first.

17. The method of Claim 12, wherein the first element is a first partial element having a related second partial element established in a second register, and the lower-level code causes the first and second partial elements to be combined after processing.

18. The method of Claim 12, wherein the act of determining first and second precisions includes determining the precisions such that the maximum negative number that can be represented in an element is one larger than the maximum negative number that can be represented in the respective precision.

19. The computer system of Claim 2, wherein the compiler generates instructions to pack multiple data elements from respective data sources into a common register to be operated on by an algorithm simultaneously with each other.

20. The computer system of Claim 19, wherein the first element is a first partial element having a related second partial element established in a second register, and the lower-level code output by the compiler causes the first and second partial elements to be combined after processing.

21. The method of Claim 11, wherein the compiler directive, instructions, or configuration file embodies instructions to compile predetermined portions of code received by the compiler to be executed simultaneously on packed data.

22. The computer program device of Claim 7, further comprising:

means for indicating whether a precision should be checked;

means responsive to the means for indicating for checking that the packed elements do not overflow or underflow or achieve a maximum magnitude negative number representable in the precision; and

means for, when packed elements overflow or underflow or achieve a maximum magnitude negative number representable in the precision in a cycle, undertaking wrap or saturation in the elements to prevent corruption of other data elements in a register, or signalling an error to be handled by an error-handling routine in the program.

23. The computer system of Claim 4, wherein the compiler determines the output precision based at least in part on an input precision.